Rule WLM012: A server defaulted to the SYSSTC Service Class

Finding: CPExpert noticed that a "server" address space defaulted to the SYSSTC Service Class.

Impact: This finding should be viewed as generally having a LOW IMPACT on the performance of the workload involved. However, the finding could have a HIGH IMPACT on performance during start-up of the server(s). Additionally, the finding could have a HIGH IMPACT on accounting for the use of system resources, on billing for use of system resources, or on capacity planning efforts.

Logic flow: This a basic finding. There are no predecessor rules.

Discussion: If subsystems are installed which support Workload Manager reporting (e.g., CICS/ESA Version 4.1 or IMS/ESA Version 5), installations can define service classes which describe particular transaction types and specify performance goals for the transactions in the service class. All transactions entering the system which fall into the workload category described by the service class are associated with the service class.

For example, an installation may wish to group all CICS transactions relating to personnel matters into a CICSPERS Service Class. The installation would define classification rules to the Workload Manager so all transactions relating to personnel matters would be placed into the CICSPERS Service Class. The installation would specify a performance goal for the CICSPERS Service Class, and an importance level for the goal.

The CICS region would report transaction performance information to the Workload Manager, and the Workload Manager would attempt to manage system resources to meet the performance goal specified for the CICSPERS Service Class.

The controlling address space must be in its own service class. In our example, suppose that the CICS region is placed into the CICSRGN Service Class. The CICSRGN Service Class would be considered a "server" and the CICSPERS Service Class might be one of several "served" service classes controlled by the CICSRGN Service Class (other CICS service classes "served" by the CICSRGN "server" might be related procurement, administration, miscellaneous, etc.).

The CICSRGN will have its own performance goals and importance. However, these performance goals and importance normally are used by the Workload Manager **only at address space start-up** time. After the CICS region has started, its performance goals and importance are ignored by the Workload Manager[1]. The Workload Manager will allocate resources based upon the performance goals and importance of the "served" service classes (in our example, the allocation will be based upon the performance of the CICSPERS transactions, and other "served" service classes served by the CICSRGN Service Class).

It is important to appreciate that the Workload Manager **does not** allocate resources to the CICSPERS Service Class, as CICSPERS is simply a logical entity which describes transactions and CICSPERS is not an address space. Rather, the Workload Manager allocates resources to the "server" address space (the CICSRGN Service Class). Similarly, the Workload Manager does not measure resources consumed by the CICSPERS Service Class, as CICS/ESA Version 4.1 does not report this information to the Workload Manager.

One implication of the structure of the "server" and "served" service classes is that the Workload Manager will attempt to meet the performance goals of all "served" service classes which are served by the "server" service class. It does this by allocating resources to the "server" service class.

From an internal view, the Workload Manager establishes *internal service classes* for server service classes. These internal service classes are named **$SRMSnnn**, where "nnn" is simply an internal counter incremented based on the number of unique internal server service classes. The Workload Manager creates an internal $SRMSnnn service class for each unique set of address spaces that serve the same set of served service classes. It is to these internal $SRMSnnn service classes that the Workload Manager assigns resources, based on the performance of the served service classes.

For example, one or more CICS regions would be described by a single internal $SRMSnnn service class, if the CICS regions provide service to the same set of served service classes. To illustrate, a CICS Terminal Owning Region (TOR) and a CICS Application Owning Region (AOR) may provide service to one or more service classes which describe CICS transactions.

The Workload Manager will make resource allocation decisions for the TOR and AOR based on the performance of the CICS transactions. The

[1]This statement is not **strictly** true. If the CICS region should become idle for an extended period (no transactions executed in the "served" service classes), the Workload Manager would use the service goal and importance specified for the CICS region service class to manage the region. Practically, of course, there would be little to manage with an idle region.

TOR and AOR address spaces will be assigned to an internal $SRMSnnn service class, and the resource allocation decisions would be made relative to the $SRMSnnn service class.

In our example, the TOR and AOR regions may be not be explicitly described by the workload classification scheme. In this case (assuming that the regions are started as started tasks), the TOR and AOR would be assigned by default to the SYSSTC Service Class, from an external view. The TOR and AOR would be managed internally by the $SRMSnnn service class as described above.

The resources used by the TOR and AOR will be combined with other started tasks executing in the SYSSTC Service Class, and reported in SMF Type 72 (Subtype 3) records. Once the resources used by the TOR and AOR are combined with other started tasks into the SYSSTC Service Class, it is impossible to account for the resources used by the TOR and AOR. Consequently, accounting and capacity planning efforts may be unsuccessful.

During the start-up of the server address spaces, the servers would be assigned the dispatching priority of the SYSSTC service class. The dispatching priority of the SYSSTC service class is very high. Start-up of the server address spaces could use a relatively significant amount of CPU time, executing at the high SYSSTC dispatching priority[2]. Consequently, performance of other service classes could be seriously degraded during start-up of the server address spaces.

CPExpert detects whether the SYSSTC Service Class provides service to any service classes. This detection is done by determining whether any elements appear in the "Service Class Served Data Section" for the SYSSTC Service Class.

CPExpert produces Rule WLM012 if any elements appear in the "Service Class Served Data Section" for the SYSSTC Service Class.

It is possible, of course, to assign the TOR and AOR to *report classes* in the workload classification scheme. In this case, the resources used by the TOR and AOR are identified and reported by the report class to which they are assigned. Unfortunately, CPExpert cannot determine whether server service classes have been assigned to report classes.

The following example illustrates the output from Rule WLM012:

[2]Address spaces in SYSSTC service class execute at dispatching priority FD (253) if APAR OW19265 is **not** applied, and execute at dispatching priority of FE (254) if OW19265 **is** applied.


```
RULE WLM012:  A SERVER DEFAULTED TO THE SYSSTC SERVICE CLASS

  CPExpert noticed that a "server" address space defaulted to the SYSSTC
  service class.  Address spaces in the SYSSTC service execute at a very
  high dispatching priority (e.g., 253).  Performance of other service
  classes could be significantly degraded during start-up of the server
  service class, as a relatively large amount of CPU time could be  used
  at a high dispatching priority.  A server address space (a CICS region,
  IMS region, etc.) normally should be explicitly assigned to a service
  class.
```

Suggestion: CPExpert suggests that you review the workload classification scheme to determine which "server" address spaces have been allowed to be assigned by default to the SYSSTC Service Class. CPExpert suggests that the workload classification scheme be adjusted to assign the "server" address spaces (1) to their own service class or (2) to a report class. There are several significant reasons for this suggestion:

- Performance of other service classes could be seriously degraded during start-up of the server address spaces.

- Storage isolation is not invoked for SYSSTC started tasks unless cross memory page faults are affecting other work with goals. Storage isolation can improve the performance of servers (e.g., CICS or IMS) at many sites.

- Management reporting will be misleading if the resources used by the servers are grouped into the SYSSTC Service Class.

- Resource accounting and billing will be unable to determine the resources used by the servers and thus will be unable to allocate resource use to their corresponding transactions. Consequently, resource accounting and billing will be unable to apportion the resource use to organization entities or applications using the resources.

- Capacity planning efforts will not be able to identify the resources used by the type of transactions, and consequently capacity planners will have insufficient information.

Reference: MVS Planning: Workload Management

MVS/ESA(SP 5):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V1R1):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V1R2):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V1R3):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R4):	Chapter 8: Defining Service Classes and Performance Goals

OS/390 (V2R5):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R6):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R7):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R8):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R9):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R10):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R1):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R2):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R3):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R4):	Chapter 8: Defining Service Classes and Performance Goals |

MVS Programming: Workload Management Services

MVS/ESA(SP 5):	Chapter 4: Using SMF Record Type 99
OS/390 (V1R1):	Chapter 7: Using SMF Record Type 99
OS/390 (V1R2):	Chapter 7: Using SMF Record Type 99
OS/390 (V1R3):	Chapter 9: Using SMF Record Type 99
OS/390 (V2R4):	Chapter 9: Using SMF Record Type 99
OS/390 (V2R5):	Chapter 10: Using SMF Record Type 99
OS/390 (V2R6):	Chapter 10: Using SMF Record Type 99
OS/390 (V2R7):	Chapter 10: Using SMF Record Type 99
OS/390 (V2R8):	Chapter 10: Using SMF Record Type 99
OS/390 (V2R9):	Chapter 10: Using SMF Record Type 99
OS/390 (V2R10):	Chapter 10: Using SMF Record Type 99
z/OS (V1R1):	Chapter 10: Using SMF Record Type 99
z/OS (V1R2):	Chapter 10: Using SMF Record Type 99
z/OS (V1R3):	Chapter 10: Using SMF Record Type 99
z/OS (V1R4):	Chapter 10: Using SMF Record Type 99 |

“WLM Server Management”, Peter Enrico, SHARE 1999 Summer Conference.

